

TW

**stichting
mathematisch
centrum**



TW

AFDELING TOEGEPASTE WISKUNDE

TN 69/72

SEPTEMBER

M. BAKKER
EEN ALGOL 60 PROGRAMMA VOOR EEN RUNGE-KUTTA-SCHEMA
VAN VARIABELE ORDE EN GRAAD

2e boerhaavestraat 49 amsterdam

Printed at the Mathematical Centre, 49, 2e Boerhaavestraat, Amsterdam.

The Mathematical Centre, founded the 11-th of February 1946, is a non-profit institution aiming at the promotion of pure mathematics and its applications. It is sponsored by the Netherlands Government through the Netherlands Organization for the Advancement of Pure Research (Z.W.O), by the Municipality of Amsterdam, by the University of Amsterdam, by the Free University at Amsterdam, and by industries.

Inhoud

Hoofdstuk	bladzijde
1. Inleiding	2
2. Gestabiliseerde RK-schema's	4
2.1 Structuur van het RK-schema	4
2.2 Bepaling van graad en orde van het RK-schema	7
2.3 De elementen van de karakteristieke matrix	10
2.4 Formules voor een schatting van de lokale fout	11
3. De procedure ordervarying runge kutta	15
3.1 Heading en parameters	15
3.2 De procedure body	17
3.3 De procedure normfunction	21
3.4 De procedure local error bound	21
3.5 De procedure local error construction	21
3.6 De procedure stepsize	21
3.7 De procedure coefficient	22
3.8 De procedure difference scheme	22
3.9 De procedure vecvec	22
4. Testresultaten	23
4.1 Een eenvoudige niet-lineaire differentiaalvergelijking	24
4.2 De Van de Pol vergelijking	26
4.3 De vergelijking van Robertson	28
4.4 Een niet-autonome differentiaalvergelijking	31
Literatuur	34

1. Inleiding

Runge-Kutta-schema's worden gebruikt voor het numeriek oplossen van beginwaardeproblemen van (vektor-) differentiaalvergelijkingen van de eerste orde.

Een dergelijk beginwaardeprobleem is van de vorm

$$(1.1) \quad \left\{ \begin{array}{l} \frac{d}{dt} \vec{U} = \vec{F}(t, \vec{U}), \quad t_0 \leq t \leq t_1 \\ \vec{U} = \vec{U}_0, \quad t = t_0. \end{array} \right.$$

Een Runge-Kutta-schema wordt vooral gebruikt, als de hogere afgeleiden van \vec{U} moeilijk zijn te berekenen, of waarvan het zelfs onmogelijk is ze te evalueren.

In deze technische notitie wordt een methode behandeld, waarbij de coëfficiënten van het Runge-Kutta-schema zodanig worden bepaald, dat de oplossing van (1.1) niet nodeloos te nauwkeurig wordt berekend als gevolg van strenge stabiliteitseisen. Deze methode houdt in, dat de coëfficiënten van het Runge-Kutta-schema (voortaan ook wel aangeduid met RK-schema) kontinu afhangen van de grootte $\sigma(D)$, waarbij σ de - door nauwkeurigheidsoverwegingen bepaalde - staplengte is en $\sigma(D)$ spektraalradius van de Jacobiaan

$$D = \left(\frac{\partial F_i}{\partial U_j} \right).$$

Het in hoofdstuk 2 behandelde RK-schema is numeriek stabiel, als de eigenwaarden van D (bijna) negatief zijn of in het linkerhalfvlak binnen een cirkel met straal $\sqrt{3}$ liggen. In onze theoretische beschouwingen gaan we ervan uit, dat de eigenwaarden van D negatief zijn. Een voordeel van het in deze technische notitie beschreven RK-schema is, zoals al werd vermeld, dat de oplossing van (1.1) niet onnodig nauwkeurig wordt berekend door stabiliteitseisen. Een nadeel is dat bij iedere integratiestap de coëfficiënten van het RK-schema opnieuw moeten worden berekend.

In hoofdstuk 2 worden de theoretische achtergronden van de procedure ordervarying runge kutta besproken, in het bijzonder het bij

deze methode behorende karakteristieke polynoom.

In hoofdstuk 3 wordt de ALGOL 60-tekst van de procedure order-varying runge kutta gegeven en worden de subprocedures besproken.

In hoofdstuk 4 worden vier testresultaten behandeld.

Tenslotte wil de auteur de heer P.A. Beentjes bedanken, wiens procedure modified runge kutta voor een deel kon worden overgenomen.

2. Gestabiliseerde RK-schema's

In dit hoofdstuk behandelen we beknopt de theoretische achtergronden van de procedure ordervarying runge kutta. Voor een volledige beschouwing van gestabiliseerde RK-schema's verwijzen we naar [1] en [2].

2.1. Struktuur van het RK-schema

De oplossing van het beginwaardeprobleem

$$(2.1) \quad \left\{ \begin{array}{l} \frac{d}{dt} U = F(t, U) , \quad t_0 \leq t \leq T , \\ U = U_0 , \quad t = t_0 , \end{array} \right.$$

waarbij we de vektorpijlen voortaan weglaten, kan in de punten $t_0, t_1, \dots, t_N = T$ benaderd worden door de vektoren u_k ($k=0,1,\dots,N$), die gegenereerd worden door het n -punts RK-schema

$$(2.2) \quad \left\{ \begin{array}{l} u_0 = U_0 , \\ u_{k+1} = u_k + \theta_0 r_k^{(0)} + \dots + \theta_{n-1} r_k^{(n-1)} , \\ r_k^{(0)} = \tau_k F(t_k, u_k) , \\ r_k^{(j)} = \tau_k F(t_k + \mu_j \tau_k, u_k + \sum_{l=0}^{j-1} \lambda_{jl} r_k^{(l)}) , \\ \tau_k = t_{k+1} - t_k , \\ k = 0, \dots, N-1, \quad j = 1, \dots, n-1 , \end{array} \right.$$

waarbij μ_j en λ_{jl} ($j,1,\dots,n-1, l=0,\dots,j-1$)

nog onbepaalde koëfficiënten zijn. In het vervolg laten we de index k van τ_k steeds weg.

Schema (2.2) kan worden gekarakteriseerd door de $n \times n$ matrix

$$(2.3) \quad R = \begin{bmatrix} \mu_1 & \lambda_{1,0} & 0 & \cdot & \cdot & \cdot & 0 \\ \cdot & \cdot & \lambda_{2,1} & \cdot & & & \cdot \\ \cdot & \cdot & & & \cdot & & \cdot \\ \cdot & \cdot & & \cdot & & & \cdot \\ \mu_{n-1} & \lambda_{n-1,0} & \lambda_{n-1,1} & \cdot & \cdot & \cdot & \lambda_{n-1,n-2} \\ \theta_0 & \theta_1 & \theta_2 & & & & \theta_{n-1} \end{bmatrix}.$$

De oplossing van (2.1) kan evenwel ook lokaal worden benaderd door de formule

$$(2.4) \quad u_{k+1} = P_n(\tau D)u_k + \text{korrektieterm},$$

waarin

$$D = \left(\frac{\partial F_i}{\partial U_j} \right) \quad t = t_k, \quad U = u_k$$

en

$$(2.5) \quad P_n(x) = 1 + \beta_1 x + \dots + \beta_n x^n$$

een reëel polynoom van de graad n is.

$P_n(x)$ heet ook wel het genererend of karakteristiek polynoom.

Indien

$$\beta_i = \frac{1}{i!}, \quad i = 1, \dots, p,$$

waarbij p een vast natuurlijk getal is ($p \leq n$), dan heet (2.4) een p^e orde nauwkeurige benadering van de oplossing van (2.1) en wel, omdat

$$(2.6) \quad ||U(t_{k+1}) - u_{k+1}|| = O(\tau^{p+1}).$$

Indien (2.6) ook voor het schema (2.2) geldt, dan heet (2.2) p^e orde consistent. Tussen de coëfficiënten β_i ($i=1, \dots, n$) en de elementen van de matrix (2.3) bestaat een verband, dat in [2] wordt gegeven door middel van een stelsel niet-lineaire vergelijkingen. Aangezien dit

stelsel onderbepaald is, kan schema (2.2) aanzienlijk vereenvoudigd worden door geschikt gekozen elementen van de matrix (2.3) nul te stellen. In het geval $p = 1, 2, 3$ kan schema (2.2) vereenvoudigd worden tot

$$(2.7) \left\{ \begin{array}{lcl} u_0 & = & U_0 \\ u_{k+1} & = & u_k + \theta_0 r_k^{(0)} + \theta_{n-1} r_k^{(n-1)} \\ r_k^{(0)} & = & \tau F(t_k, u_k) \\ r_k^{(j)} & = & \tau F(t_k + \mu_j \tau, u_k + \lambda_{j,0} r_k^{(0)} + \lambda_{j,j-1} r_k^{(j-1)}) \\ & & k = 0, \dots, N-1, \quad j = 1, \dots, n-1 \end{array} \right. ,$$

Het verband tussen de coëfficiënten van (2.7) en $\beta_i (i=1, \dots, n)$ is nu

$$(2.8) \left\{ \begin{array}{lcl} \theta_{n-1} & = & \begin{cases} \frac{3}{4} & , \text{ als } p = 3 \\ 1 & , \text{ als } p = 1, 2 \end{cases} \\ \theta_0 & = & 1 - \theta_{n-1} \\ \lambda_{j,0} & = & \theta_0, \quad j = 2, \dots, n-1 \\ \lambda_{n-1,n-2} & = & \frac{\beta_2}{\theta_{n-1}} - \theta_0 \\ \lambda_{j,j-1} & = & \frac{\beta_{n-j+1}}{\theta_{n-1} \prod_{k=j+1}^{n-1} \lambda_{k,k-1}} - \theta_0, \quad j = 2, \dots, n-2 \\ \lambda_{1,0} & = & \frac{\beta_n}{\theta_{n-1} \prod_{k=2}^{n-1} \lambda_{k,k-1}} \\ \mu_1 & = & \lambda_{1,0} \\ \mu_j & = & \lambda_{j,j-1} + \theta_0, \quad j = 2, \dots, n-2 \end{array} \right. ,$$

In het geval $p = 1, 2$ wordt dit gereduceerd tot

$$(2.8a) \quad \left\{ \begin{array}{lll} \theta_0 & = 0, \theta_{n-1} = 1 & , \\ \lambda_{n-1,n-2} & = \beta_2 & , \\ \lambda_{j,j-1} & = \frac{\beta_{n-j+1}}{\beta_{n-j}} & , \\ \mu_j & = \lambda_{j,j-1} & , \quad j = 1, \dots, n-1, \\ \lambda_{j,0} & = \theta_0 = 0 & , \quad j = 2, \dots, n-1 \end{array} \right.$$

2.2. Bepaling van graad en orde van het RK-schema

Zoals in [2] wordt aangetoond is schema (2.4) en daarmee schema (2.7) numeriek stabiel, als

$$(2.9) \quad |P_n(\tau \delta_j)| \leq 1 ,$$

waarbij δ_j de eigenwaarden van de Jacobiaan D zijn. Deze ongelijkheid wordt in elk geval bereikt, als de eigenwaarden bijna negatief zijn.

Zoals we al eerder vermeldden, beperken we ons tot het geval, dat de eigenwaarden niet-positief reëel zijn. Dan is er numerieke stabiliteit, als

$$(2.10) \quad \tau \leq \frac{\beta(n)}{\sigma(D)} ,$$

waarbij $\beta(n)$ een positief getal is, zodanig dat geldt

$$(2.11) \quad |P_n(x)| \leq 1, \quad -\beta(n) \leq x \leq 0 .$$

We willen nu bij gegeven τ - τ is door nauwkeurigheidsoverwegingen bepaald, zie [3, blz.15] - en $\sigma(D)$ het polynoom (2.5) zodanig bepalen, dat het daaruit voortkomende schema (2.7) aan de volgende voorwaarden voldoet:

- (2.12) (a) het is numeriek stabiel ;
(b) de orde van consistentie p ($1 \leq p \leq 3$) is zo hoog mogelijk, terwijl de graad n ($n \geq 3$) zo laag mogelijk is;

$$(c) \quad \left| \frac{1}{(p+1)!} - \beta_{p+1} \right| = \text{minimum.}$$

Volgens [1, blz.20] kunnen we in het geval $\tau\sigma(D) \leq 18$ n gelijk aan 3 stellen. De koëfficiënten van $P_3(x)$ zijn dan als volgt bepaald:

Tabel 2.1. Koëfficiënten van $P_3(x)$

Gebied van $\tau\sigma(D)$	β_1	β_2	β_3
$[0, 2.52]$	1	$\frac{1}{2}$	$\frac{1}{6}$
$[2.52, 6.26]$	1	$\frac{1}{2}$	$\frac{[\tau\sigma(D)]^2 - 2\tau\sigma(D) + 4}{2[\tau\sigma(D)]^3}$
$[6.26, 18]$	1	$\frac{2}{\tau\sigma(D)} \left(1 + \sqrt{\frac{2}{\tau\sigma(D)}} \right)$	$\frac{1}{4} \beta_2^2$

Als nu echter $\tau\sigma(D) > 18$, dan is het noodzakelijk de graad van schema (2.7) en daarmee de graad van het polynoom $P_n(x)$ te verhogen, willen we een eerste orde consistent RK-schema handhaven. Aangezien het dan niet meer mogelijk is om aan alle voorwaarden van (2.12) te voldoen, moeten we genoegen nemen met een polynoom, dat een schema (2.7) genereert, dat slechts aan de voorwaarden (a) en (b) van (2.12) voldoet. Ter compensatie kunnen we dan een polynoom nemen, waarvan de koëfficiënten gemakkelijk door de rekenmachine te berekenen zijn, bv. door middel van een eenvoudige rekurrente betrekking. Teneinde een dergelijk polynoom te konstrueren, bewijzen we de volgende

Stelling : Het polynoom

$$(2.13) \quad P_n(x) = R_n^{(\alpha, \alpha)} \left(1 + \frac{2x}{\tau\sigma(D)} \right), \quad -\tau\sigma(D) \leq x \leq 0,$$

waarbij $R_n^{(\alpha, \alpha)}$ het genormaliseerde Jacobipolynoom [5, blz.1] is, genereert een numeriek stabiel en eerste orde consistent RK-schema (2.7), als aan de volgende voorwaarden is voldaan:

$$(2.14) \quad \left\{ \begin{array}{ll} (a) & n \geq \sqrt{\frac{\tau\sigma(D)}{2}} , \\ (b) & \alpha = \frac{n(n+1) - \tau\sigma(D)}{\tau\sigma(D) - 2n} . \end{array} \right.$$

Bewijs: Aangezien

$$\left| 1 + \frac{2x}{\tau\sigma(D)} \right| \leq 1 , \quad -\tau\sigma(D) \leq x \leq 0 ,$$

geldt er [5, blz.2]

$$\left| R_n^{(\alpha, \alpha)} \left(1 + \frac{2x}{\tau\sigma(D)} \right) \right| \leq 1 , \quad -\tau\sigma(D) \leq x \leq 0 ,$$

$$\alpha \geq -\frac{1}{2} ,$$

zodat voor $\alpha \geq -\frac{1}{2}$ het polynoom (2.13) een numeriek stabiel RK-schema (2.7) genereert.

Dit schema is eerste orde consistent, als

$$\left. \frac{d}{dx} P_n(x) \right|_{x=0} = 1 ,$$

dus als

$$\frac{2}{\tau\sigma(D)} \left. \frac{d}{dy} R_n^{(\alpha, \alpha)}(y) \right|_{y=1} = 1 .$$

Dit laatste houdt in, dat [6, blz.2]

$$\frac{n(n+2\alpha+1)}{(\alpha+1)\tau\sigma(D)} = 1 ,$$

waaruit weer volgt, dat

$$\alpha = \frac{n(n+1) - \tau\sigma(D)}{\tau\sigma(D) - 2n} .$$

Uit de eis $\alpha \geq -\frac{1}{2}$ volgt dan, dat

$$n \geq \sqrt{\frac{\tau\sigma(D)}{2}} ,$$

zodat bewezen is, dat het polynoom (2.13) een eerste orde consistent en numeriek stabiel RK-schema genereert, als α en n aan (2.14) voldoen.

We gebruiken dus voor schema (2.7) het karakteristieke polynoom (2.13) met

$$(2.14a) \quad \begin{cases} n = - \left[-\sqrt{\frac{\tau\sigma(D)}{2}} \right] , \\ \alpha = \frac{n(n+1) - \tau\sigma(D)}{\tau\sigma(D) - 2n} . \end{cases}$$

De koëfficiënten van dit polynoom voldoen aan de rekurrente betrekking

$$(2.15) \quad \begin{cases} \beta_1 = 1 \\ \beta_{i+1} = \beta_i \frac{(n-i)(n+2\alpha+i+1)}{(i+1)\tau\sigma(D)} , i = 1, \dots, n-1. \end{cases}$$

2.3. De elementen van de karakteristieke matrix.

Nu we aan de hand van τ en $\sigma(D)$ de graad en orde van consistentie van schema (2.7) hebben bepaald en beschikken over de koëfficiënten van het genererende polynoom, kunnen we voor de verschillende gevallen van n en p de formules (2.8) en (2.8a) uitwerken.

$$\underline{p = 3 , n = 3}$$

In dat geval is $\beta_1 = 1$, $\beta_2 = \frac{1}{2}$ en $\beta_3 = \frac{1}{6}$.

Formule (2.8) wordt dan

$$(2.16) \quad \begin{cases} \lambda_{1,0} = \frac{8}{15} , & \lambda_{2,0} = \frac{1}{4} , & \lambda_{2,1} = \frac{5}{12} , \\ \mu_1 = \frac{8}{15} , & \mu_2 = \frac{2}{3} , \\ \theta_2 = \frac{3}{4} , & \theta_0 = \frac{1}{4} . \end{cases}$$

$$\underline{p = 2 , n = 3}$$

We vullen de waarden voor β_1 , β_2 en β_3 uit tabel 2.1 in formule (2.8a) in en krijgen

$$(2.17) \quad \left\{ \begin{array}{l} \lambda_{1,0} = \mu_1 = \frac{\beta_3}{\beta_2} = \frac{4[\tau\sigma(D)]^2 - 2[\tau\sigma(D)] + 1}{[\tau\sigma(D)]^3} , \\ \lambda_{2,1} = \mu_1 = \frac{\beta_2}{\beta_1} = \frac{1}{2} , \\ \theta_2 = 1 , \quad \lambda_{2,0} = \theta_0 = 0 . \end{array} \right.$$

$$\underline{p = 1 , \quad n = 3}$$

Volgens tabel 2.1 geldt voor β_2 en β_3

$$\beta_2 = \frac{2}{\tau\sigma(D)} \left(1 + \sqrt{\frac{2}{\tau\sigma(D)}} \right) , \quad \beta_3 = \frac{1}{4} \beta_2^2 .$$

Werken we nu (2.8a) uit, dan krijgen we

$$(2.18) \quad \left\{ \begin{array}{l} \lambda_{1,0} = \mu_1 = \frac{\beta_3}{\beta_2} = \frac{1}{4} \beta_2 , \\ \lambda_{2,1} = \mu_2 = \frac{\beta_2}{\beta_1} = \beta_2 , \\ \theta_2 = 1 , \quad \lambda_{2,0} = \theta_0 = 0 . \end{array} \right.$$

$$\underline{p = 1 , \quad n = 4, 5, \dots, 10}$$

Toepassing van (2.15) geeft nu

$$(2.19) \quad \left\{ \begin{array}{l} \lambda_{n-j,n-j-1} = \mu_{n-j} = \frac{\beta_{j+1}}{\beta_j} = \frac{(n-j)(n+2\alpha+j+1)}{(j+1)\tau\sigma(D)} , \quad j = 1, \dots, n-1, \\ \theta_{n-1} = 1 , \quad \lambda_{j,0} = \theta_0 = 0 , \quad j = 2, \dots, n-1, \end{array} \right.$$

waarbij α door (2.14) gegeven is.

2.4. Formules voor een schatting van de lokale fout

De lokale fout wordt geschat volgens de formule

$$(2.20) \quad \rho_k \simeq \theta_0' r_k^{(0)} + \dots + \theta_{n-1}' r_k^{(n-1)} ,$$

waarbij $\theta'_0, \dots, \theta'_{n-1}$ zodanig bepaalde parameters zijn, dat (2.20) overeenkomt met de schatting

$$(2.21) \quad \rho_k \approx \frac{1}{2} \tau^2 \left. \frac{d^2 U_k}{dt^2} \right|_{t=t_k} + O(\tau^3).$$

We kiezen de - pessimistische - schatting (2.21), omdat dan bij wisselende orde van consistentie een zekere continuïteit in de foutenschatting en daarmee in de stapkeuze strategie [3, blz.15] wordt gegarandeerd.

Voor de gewichten $\theta'_0, \dots, \theta'_{n-1}$ houdt de compatibiliteit van (2.20) met (2.21) in, dat ze aan de volgende relaties voldoen

$$(2.22) \quad \begin{cases} \theta'_0 + \dots + \theta'_{n-1} = 0, \\ \mu_1 \theta'_1 + \dots + \mu_{n-1} \theta'_{n-1} = \frac{1}{2}. \end{cases}$$

Nu geldt voor een willekeurige norm van ρ_k de ongelijkheid

$$||\rho_k|| = ||\sum_{j=0}^{n-1} \theta'_j r_k^{(j)}|| \leq \sum_{j=0}^{n-1} |\theta'_j| ||r_k^{(j)}||$$

Uit deze ongelijkheid volgen de ongelijkheden

$$||\rho_k|| \leq \bar{\theta}' \sum_{j=0}^{n-1} ||r_k^{(j)}||$$

en

$$||\rho_k|| \leq \overline{||r_k||} \sum_{j=0}^{n-1} |\theta'_j|,$$

waarbij

$$\bar{\theta}' = \max_{j=0, \dots, n-1} |\theta'_j|$$

en

$$\overline{||r_k||} = \max_{j=0, \dots, n-1} ||r_k^{(j)}||.$$

Het ligt dus voor de hand om aan het onderbepaalde stelsel (2.22) een extra eis toe te voegen, namelijk [6,6]

$$(2.23) \quad \max_{j=0, \dots, n-1} |\theta'_j| = \text{minimum}$$

of

$$(2.24) \quad \sum_{j=0}^{n-1} |\theta'_j| = \text{minimum}$$

Het minimaxprobleem (2.23) met nevenvoorwaarden (2.22) is bijzonder lastig op te lossen. Weliswaar kan (2.23) min of meer benaderd worden door het kleinstekwadratenprobleem.

$$\sum_{j=0}^{n-1} |\theta'_j|^2 = \text{minimum},$$

maar de oplossing daarvan, te weten

$$\left\{ \begin{array}{ll} \theta'_j = \alpha + \beta \mu_j, & j = 1, \dots, n-1, \\ \theta'_0 = \alpha & , \\ \alpha = -\frac{1}{2} \frac{A}{nB - A^2} & , \\ \beta = \frac{1}{2} \frac{n}{nB - A^2} & , \\ A = \sum_{j=1}^{n-1} \mu_j & , \\ B = \sum_{j=1}^{n-1} \mu_j^2 & \end{array} \right.$$

vergt veel rekentijd, wat extra nadelig is, omdat $\theta'_0, \dots, \theta'_{n-1}$ iedere integratiestap opnieuw uitgerekend moeten worden.

(2.24) met nevenvoorwaarden (2.22) is een probleem uit de lineaire programmering. Het heeft in principe als oplossing

$$\theta'_k, \theta'_1 \neq 0, \quad \theta'_j = 0, \quad j \neq k, 1,$$

voor zekere k en 1 . Stellen we nu even $\mu_0 = 0$, dan is de oplossing van (2.22)

$$\theta'_k = -\theta'_1 = \frac{1}{2(\mu_1 - \mu_k)}.$$

Vullen we dit in (2.24) in, dan vinden we dat we een k en een 1 moeten zoeken, waarvoor

$$(2.24a) \quad \frac{1}{|\mu_k - \mu_1|} = \text{minimum}.$$

Slaan we de formules (2.16), ..., (2.19) hierop na, dan vinden we dat aan (2.24a) voldaan wordt, als

$$k = n - 1, \quad 1 = 0.$$

De oplossing van (2.24) met nevenvoorwaarden (2.22) is dus

$$(2.25) \quad -\theta'_0 = \theta'_{n-1} = \frac{1}{2\mu_{n-1}}, \quad \theta'_j = 0, \quad j = 1, \dots, n-2.$$

We schatten de lokale fout dus volgens de formule

$$(2.26) \quad \rho_k \approx \frac{1}{2\mu_{n-1}} [r_k^{(n-1)} - r_k^{(0)}].$$

3. De procedure ordervarying runge kutta

In dit hoofdstuk bespreken we de procedure ordervarying runge kutta en zijn subprocedures.

3.1 Heading en parameters

De heading van de procedure is

```
procedure ordervarying runge kutta (t, te, m0, m, u, sigma, deriva  
tive, k, alfa, norm, aeta, reta, eta, rho, output);  
integer m0, m, k, norm;  
real t, te, sigma, alfa, aeta, reta, eta, rho;  
array u;  
procedure derivative, output;
```

Parameters:

t	:	<variable> ;
		t wordt gebruikt als Jensenparameter; bij een aanroep van ordervarying runge kutta moet t de beginwaarde t_0 van de onafhankelijke variabele hebben;
te	:	<expression> ;
		de eindwaarde van t;
m0, m	:	<expression> ;
		indices van de eerste en laatste vergelijking van het op te lossen stelsel;
u	:	<array identifier> ;
		de dimensie van u is 1; bij een aanroep van ordervarying runge kutta moet het array de waarden van $u(t_0)$ hebben;

sigma : <expression> ;
 spektraalradius van de Jacobiaan van de differentiaalvergelijking; sigma moet door de gebruiker worden gegeven; als de Jacobiaan complexe eigenwaarden heeft, moet sigma de waarde 0 krijgen;

derivative : <procedure identifier> ;
 derivative moet als volgt door de gebruiker worden gegeven:
procedure derivative (x,a);
real x; array a;
 <vervanging van a_i door
 $F_i(x, a_{m0}, \dots, a_m) (i=m0, \dots, m)>$;

k : <variable> ;
 telt het aantal integratiestappen;
 bij de eerste aanroep van ordervarying runge
 kutta moet k gelijk aan 0 zijn;

alfa : <expression> ;
 deze parameter laat τ_k aan de ongelijkheid

$$\tau_k \leq \alpha \frac{\tau_{k-1} \sigma_{k-1}}{\sigma_k}$$

voldoen, bij voorkeur dient α tussen 1 en 2 te liggen;

norm : <expression> ;
 voor norm = 1 wordt met de maximumnorm,
 voor norm = 2 met de euklidische norm gerekend;

aeta, reta : <expression> ;
 verlangde relatieve en absolute lokale precisie; als aeta en reta beide negatief zijn, wordt door ordervarying runge kutta niet aan nauwkeurigheidskondities voldaan, hetgeen inhoudt, dat er geen lokale fout wordt berekend en dat de staplengte voldoet aan de formule

$$\tau_k = \min \left(t - t_k, \frac{195}{\sigma_k} \right)$$

eta : <variable> ;
 de tolerantie of globale fout η_k , die een functie is van aeta, reta en $||\dot{u}(t_k)||$;

rho : <variable> ;
 de diskrepantie of lokale fout, gemaakt in de laatste integratiestap;

output : <procedure identifier> ;
 de heading van output is:
procedure output;
 door middel van deze subprocedure kan de gebruiker aan het eind van een integratiestap resultaten laten afdrukken (bv. k, t, eta, ...) of tussentijds in het integratieproces ingrijpen (bv. door het statement if k > 100 then exit) ;

3.2 De procedure body

De body van de procedure is:

```

begin integer i,p,n;
      own real ec0,ec1,ec2,tau0,tau1,tau2,taus,t2,betan;
      real theta0,thetanm1,thetaacc,tau,gamma,sigma1;
      array labda[0:9],ro,r[m0:m];
      boolean start,step1,p3,tstb;

real procedure normfunction(nrm,w);
integer nrm;array w;
begin integer j; real s,x;
  s:= if nrm = 1 then 0 else if nrm = 2 then
    sqrt(vecvec(m0,m,0,w,w)) else 1 ;
  if nrm=1 then for j:=m0 step 1 until m do
    begin x:=abs(w[j]);
      if x>s then s:=x
    end;
  normfunction:=s
end normfu;

procedure local error bound;
eta:=aeta+retaxnormfunction(norm,u);

procedure local error construction;
begin integer j; real tht;
  if i=0 then
    begin thetaacc:= .5/(labda[n-1] + theta0);
      for j:=m0 step 1 until m do ro[j]:=0
    end;
  tht:=( if i=0 then -thetaacc else + thetaacc)xtau;
  for j:=m0 step 1 until m do
    ro[j]:=ro[j]+thtXr[j];
  if i=n-1 then
    begin rho:=normfunction(norm,ro);
      ec0:=ec1;ec1:=ec2;ec2:=rho/tau^2
    end
end l.e.c.;

procedure coefficient;
begin integer j;switch order:=a1,a2,a3;
      real betanrec;
      betan:=tauXsigma1+.1;
      p:=if betan<2.52 then 3 else
        if betan<6.26 then 2 else 1;
      n:=1+(if betan<17.6 then 2 else
        entier(sqrt(betan/1.96)));
      go to if (p3^p=3)V(tstb^betan>195.0999999)
      then a4 else order[p];
a1:   if n=3 then

```

```
begin    betanrec:=2/betan;
        labda[2]:=betanrec*(1+sqrt(betanrec));
        labda[1]:=labda[2]*.25
end else
begin    real alfa,twoalfa;
        alfa:=(n*(n+1)-betan)/(betan-n-n);
        twoalfa:=alfa+alfa;
        for j:=2 step 1 until n do
            labda[n-j+1]:=(n-j+1)*(n+j+twoalfa)/
            betan/(alfa+j)/j
        end ;
        go to a4;
a2:      betanrec:=1/betan;labda[2]:=1;
        labda[1]:=betanrec*(betanrec*(betanrec*4-2)+1);
        go to a4;
a3:      labda[1]:=1.283 333 333 3333 ;
        labda[2]:=1.416 666 666 6667;
a4:      p3:=p=3;tstb:=betan>195.09999999;
        thetanm1:=if p3 then .75 else 1;
        theta0:=1-thetanm1
end coefficient;
```

```
procedure stepsize;
begin real tauacc,taustab,aa,bb,cc,ec;
    local error bound;
    sigma1:=sigma;
    if eta>0 then
    begin if start then
        begin if k=0 then
            begin integer j;
                for j:=m0 step 1 until m do
                    r[j]:=u[j];derivative(t,r);
                    tauacc:=eta/normfunction(norm,r);
                    step1:=true
                end else
                if step1 then
                begin tauacc:=sqrt(eta/rho)*tau2;
                    if tauacc>10*tau2 then
                        tauacc:=10*tau2 else step1:=false
                    end else
                begin bb:=(ec2-ec1)/tau1; cc:=ec2-bb*tau2;
                    ec:=bb*tau+cc;
                    tauacc:=if ec<0 then tau2 else sqrt(eta/ec);
                    start:=false
                end
            end else
        end
```

```

begin aa:=((ec0-ec1)/tau0+(ec2-ec1)/tau1)
/ (tau1+tau0);
bb:=(ec2-ec1)/tau1-aa*(2xt2-tau1);
cc:=ec2-t2*(bb+aa*xt2); ec:=cc+tx(bb+txaa);
tauacc:=if ec<0 then
tau else sqrt (eta/ec);
tau0:=alfaxbetan/sigma1;
iftauacc>tau0thentauacc:=tau0;
tau0:=gamma*taus;
iftauacc<tau0thentauacc:=tau0;
if tauacc<10-12 × t then tauacc:= 10-12 × t
end
end else tauacc:=te-t;
taustab:=if sigma1<10-10then1.7else195/sigma1;
iftaustab<10-12xtthen
beginoutput; goto end of ord var r kend;
tau:=if tauacc>taustab then taustab else tauacc;
taus:=tau; if tau>te-t then tau:=te-t;
tau0:=tau1;tau1:=tau2;tau2:=tau
end
stepsizes;

procedure difference scheme;
begin integer j;real mt,lt;
i:=1;
next term:
lt:=labda[i+1]*tau;
mt:=( if p3^i+1 then tau*theta0 else 0 ) + lt ;
for j:=m0 step 1 until m do r[j]:=u[j]+lt*xr[j];
i:=i+1;derivative(t+mt,r);
if eta>0^(i=0^i=n-1) then local error construction;
if (i=0^p3)^i=n-1 then
begin real tht;
tht:=if i=0 then theta0*tau else thetanm1*tau;
for j:=m0 step 1 until m do
u[j]:=u[j]+tht*xr[j]
end;
if i<n-1 then goto next term;
t2:=t;t:=t+tau
end diff.sch.;

start:=k=0;p3:=tstb:= false;
gamma:=.5;labda[0]:=0;
next level:
stepsizes;coefficient;difference scheme;
k:=k+1;output;if t<te then go to next level ;
end of ord var r k :
end order varying rk ;

```

3.3 De procedure normfunction

Deze procedure berekent voor alle één dimensionale arrays met grenzen m0 en m een norm en wel als volgt:

de maximumnorm , als norm = 1 ;

de euklidische norm , als norm = 2 .

Voor alle overige waarden van norm levert deze procedure de waarde 1 af.

3.4 De procedure local error bound

De berekening van eta volgens de formule $\eta = a\eta + r\eta * \text{normfunction}(\text{norm}, u)$.

3.5 De procedure local error construction

ρ_k wordt geschat volgens formule (2.26). Daarna wordt $||\rho_k||$ berekend als in 3.3 is aangegeven.

3.6 De procedure stepsize

Voor een uitvoerige beschrijving zie [3, 3.7]
Voor $||\rho||$ gebruiken we de representaties

$$||\rho|| = C\tau^2 ,$$

$$||\rho|| = (Bt+C)\tau^2 ,$$

en

$$||\rho|| = (At^2+Bt+C)\tau^2 .$$

Verder onderwerpen we in afwijking van [3,3.7] de stapgrootte aan de beperkingen

$$\begin{cases} \frac{1}{2}\tau_{k-1} \leq \tau_k \leq \text{alfa} * \tau_{k-1} * \frac{\sigma(D_{k-1})}{\sigma(D_k)} , \\ \tau_k \leq 195/\sigma(D_k) . \end{cases}$$

3.7 De procedure coëfficient

Deze procedure verloopt als volgt:

- 1) $\tau * \sigma(D)$ wordt berekend en aan de hand daarvan n en p ;
- 2) θ_0, θ_{n-1} en de coëfficienten $\lambda_{1,0}, \dots, \lambda_{n-1,n-2}$ werden berekend volgens 2.3; omdat μ_j ($j=1, \dots, n-1$); alleen in het geval $p = 3$ ongelijk λ_j is en dan ook nog uit twee konstante getallen bestaat is er geen array voor μ_j nodig. De expressies, die u_j bevatten worden in het geval $p = 3$ apart berekend en voor het overige gelijkgesteld aan de overeenkomstige expressies met λ_j ; verder wordt ook nog θ'_{n-1} berekend.

3.8 De procedure difference scheme

u_k wordt omgezet in u_{k+1} volgens schema (2.7). Tegelijkertijd wordt in het geval $\eta > 0$ de lokale afbreekfout door middel van de procedure local error construction berekend.

3.9 De procedure vecvec

Zie [4, blz.8] .

4. Testresultaten

De volgende problemen werden met behulp van de procedure order-varying runge kutta opgelost:

(a) het beginwaarde probleem

$$\begin{cases} \frac{du}{dt} = 100 - u^2 & , \quad t \geq 0 \\ u = 0 & , \quad t = 0 \end{cases}$$

werd over het interval $[0,6]$ geïntegreerd met 1, 2, ..., 5 als referentiepunten (dus in feite werd achtereenvolgens over de intervallen $[0,1]$, $[1,2]$, ..., $[5,6]$ geïntegreerd).

(b) de Van de Pol vergelijking

$$\begin{cases} \ddot{u} - \mu(1-u^2) \dot{u} + u = 0 & , \quad t \geq 0 , \\ u(0) = 2 , \dot{u}(0) = 0 \end{cases}$$

werd voor $\mu = 10$ over het interval $[0,18.86305053]$ geïntegreerd;

(c) de reaktievergelijking van Robertson [7,ch.6.1]

$$\begin{cases} \dot{u}_1 = -.04 u_1 + 10^4 u_2 u_3 & , \\ \dot{u}_2 = +.04 u_1 - 10^4 u_2 u_3 - 3_{10}^7 u_2^2 & , \\ \dot{u}_3 = & + 3_{10}^7 u_2^2 & , \quad t \geq 0 , \\ u_1(0) = 1 , u_2(0) = u_3(0) = 0 \end{cases}$$

werd over het interval $[0,10]$ geïntegreerd met 0.4 als referentiepunt;

(d) het niet-autonome beginwaardeprobleem

$$\begin{cases} \dot{u} = -e^t(u - \ln(t) + \frac{1}{t}) & , \quad t \geq .01 & , \\ u = -2\ln(10) \end{cases}$$

werd over het interval $[.01, 10]$ geïntegreerd.

4.1 Een eenvoudige niet-lineaire differentiaalvergelijking

Het beginwaardeprobleem

$$(4.1) \begin{cases} \dot{u} = 100 - u^2 & , \quad t \geq 0 & , \\ u = 0 & , \quad t = 0 \end{cases}$$

heeft als exakte oplossing

$$(4.2) \quad \tilde{u} = 10 \frac{e^{20t} - 1}{e^{20t} + 1} = 10 - \frac{20}{e^{20t} + 1} .$$

Aangezien voor $t \ll 1$ de afgeleide groot is en daarna snel afneemt tot 0, ligt het voor de hand dat op het interval $[0, 1]$ de oplossing van (4.1) in veel meer punten wordt berekend dan op de intervallen $[1, 2], \dots, [5, 6]$. We integreerden (4.1) over het interval $[0, 6]$ met 1, 2, 3, 4 en 5 als referentiepunten. Per interval bekijken we het aantal integratiestappen l , nodig om het interval te doorlopen, de maximale gemeten fout, dwz.

$$\epsilon_{\max} = \max_{t_k \in I} ||\tilde{u}(t_k) - u_k||$$

en de maximale geschatte fout op het interval, dwz.

$$\rho_{\max} = \max_{t_k \in I} ||\rho_k|| .$$

Voor a_{eta} en r_{eta} kozen we resp. 10^{-1} en 10^{-3} .

Tabel 4.1 Resultaten voor $\alpha = .1 + .001 * ||u||$

Interval	ρ_{\max}	ϵ_{\max}	1
[0,1]	.5490	$.1423_{10^{-1}}$	17
[1,2]	$.7735_{10^{-2}}$	$.4978_{10^{-3}}$	3
[2,3]	$.6519_{10^{-2}}$	$.3027_{10^{-3}}$	3
[3,4]	$.2416_{10^{-2}}$	$.1826_{10^{-3}}$	3
[4,5]	$.1725_{10^{-3}}$	$.3635_{10^{-4}}$	5
[5,6]	$.3063_{10^{-4}}$	$.3951_{10^{-5}}$	4

Zoals te verwachten was, werd de oplossing van (4.1) op het interval [0,1] in veel meer punten berekend dan op de overige intervallen, als gevolg van de steile helling van u in de buurt van de oorsprong. We illustreren dit aan de hand van een grafiek van de exakte oplossing van (4.1)

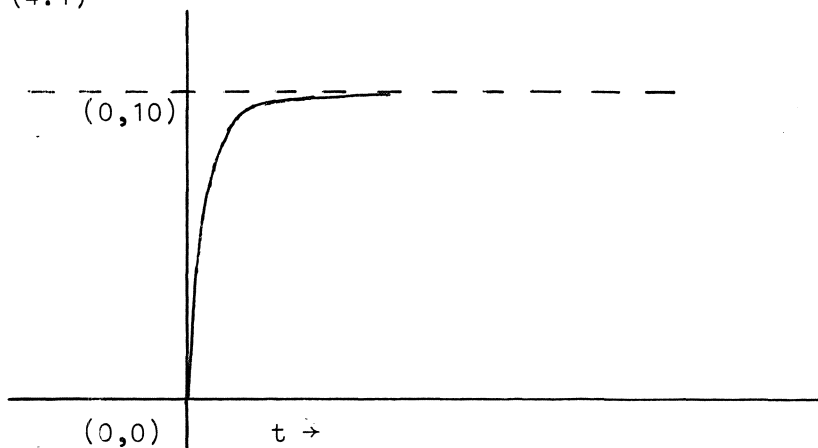


fig. 4.1 Grafiek van $\tilde{u} = 10 - \frac{20}{e^{20t} + 1}$

De graad van het RK schema (2.7) was steeds 3. De orde van nauwkeurigheid was op het hele interval $[0,1]$ gelijk aan 3, met uitzondering van de laatste twee integratiestappen. Op de overige intervallen was de orde van nauwkeurigheid 1 of 2 behalve bij de berekening van u_k in de referentiepunten 3, 4, 5, 6.

Tenslotte geven we nog de procedure derivative en de aanroep van ordervarying runge kutta

```

procedure derivative (x,a); real x; array a;

a[0]:= 100 - a[0] ↑ 2 ;

k:= 0; u[0]:= t:= 0 ;

for te:= 1, 2, 3, 4, 5, 6 do

ordervarying runge kutta (t, te, 0, 0, u, 2 * u[0] ,

derivative, k, 1.5, 1, 10-1, 10-3, eta, rho, output);

```

4.2 De Van de Pol vergelijking

We bekijken voor $\mu = 10$ het beginwaardeprobleem

$$(4.3) \quad \begin{cases} \ddot{x} + \mu(x^2 - 1) \dot{x} + x = 0 & , \quad t \geq 0 , \\ x = 2, \dot{x} = 0 & , \quad t = 0 . \end{cases}$$

(4.3) kan worden omgezet in het eerste orde beginwaardeprobleem

$$(4.4) \quad \begin{cases} \dot{x}_1 = x_2 + \mu(1 - \frac{x_1^2}{3})x_1 \\ \dot{x}_2 = -x_1 \\ x_1 = 0, x_2 = \frac{2}{3}\mu \end{cases} \quad , \quad \begin{matrix} t \geq 0 , \\ t = 0 . \end{matrix}$$

x_1 is dan de oplossing van (4.3). Terloops zij vermeld dat x_2 de oplossing is van de vergelijking van Rayleigh

$$\begin{cases} \ddot{x}_2 - \mu \dot{x}_2 + \frac{1}{3} \mu \dot{x}_2^3 + x_2 = 0 & , \quad t \geq 0 , \\ x_2 = \frac{2}{3}\mu, \dot{x}_2 = -2 & , \quad t = 0 . \end{cases}$$

We integreerden (4.4) voor verschillende waarden van $\eta = a\eta$ over het interval $[0, T]$ met $T = 18.86305053$ en vergeleken de gevonden waarden van $x_1(T)$, $x_2(T)$ en $\dot{x}_1(T)$ met die uit [3, blz.34]. Deze waarden zijn resp. 2.01428536, 7.09931864 en 0. We definieerden

$$\begin{cases} \epsilon_1 = |x_1(T) - 2.01428536|, \\ \epsilon_2 = |x_2(T) - 7.09931864|, \\ \epsilon_3 = |\dot{x}_1(T)|. \end{cases}$$

We kregen de volgende resultaten:

$\eta = a\eta$ ($r\eta = 0$)	ϵ_1	ϵ_2	ϵ_3	K
.01	2.1_{10}^{-4}	3.2_{10}^{-2}	3.9_{10}^{-2}	127
.001	3.8_{10}^{-5}	1.8_{10}^{-4}	1.0_{10}^{-3}	328
.0001	1.4_{10}^{-6}	6.1_{10}^{-6}	3.8_{10}^{-5}	1000

Tabel 4.2

K is hier het aantal integratiestappen. De graad van het RK-schema en de orde van nauwkeurigheid waren steeds 3. Voor een vergelijking van de resultaten uit tabel 4.7 met overeenkomstige resultaten, verkregen door een andere procedure zie [3, blz.19]. ϵ_1 , ϵ_2 en ϵ_3 worden daar aangeduid met $|\epsilon_{\dot{x}}|$, $|\epsilon_x|$ en $|\epsilon_{\ddot{x}}|$.

De procedure derivative was

```

procedure derivative (x,a); real x; array a;
begin real a1;
      a1:= a[1] ;
      a[1]:= a[2] + muu * (1-a1+2/3) * a1 ;
      a[2]:= -a1
end ;

```

De aanroep van ordervarying runge kutta was

```

muu:= 10;
for aeta:= 10-2, 10-3, 10-4 do
begin k:= 0; t:= 0; x[1]:= 2; x[2]:= 2 * muu/3;
      ordervarying runge kutta (t, 18.86305053, 1, 2, x, sigma, deri
      vative, K, 1.5, 2, aeta, 0, eta, rho, output)
end ;

```

sigma werd gegeven door middel van de procedure

```

real procedure sigma;
begin real d;
      d:= .5 * muu * (1-x[1]2);
      sigma:= if d < -1 then
      - d + sqrt (d*d-1) else 0
end ;

```

Tot slot merken we op, dat het effect van eigenwaarden van de Jacobiaan met een positief reëel deel niet groot was, omdat de afgeleide van x[1] dan zo groot was (de eigenwaarden zijn alleen afhankelijk van x[1]), dat de "kritieke zône" snel werd gepasseerd.

4.3 De vergelijking van Robertson

We bekijken het beginwaardeprobleem [7, ch.6.1] .

$$(4.5) \quad \begin{cases} \dot{u}_1 = - .04 u_1 + 10^4 u_2 u_3 & , \\ \dot{u}_2 = + .04 u_1 - 10^4 u_2 u_3 - 3_{10} 7 u_2^2 & , \\ \dot{u}_3 = & + 3_{10} 7 u_2^2 & , t \geq 0 , \\ u_1 = 1, u_1 = u_3 = 0 & , t = 0 . \end{cases}$$

Door optelling zien we onmiddellijk, dat

$$\frac{d}{dt} (u_1 + u_2 + u_3) = 0 ,$$

waaruit blijkt, dat

$$u_1 + u_2 + u_3 = c .$$

Uit de beginwaarden leiden we af, dat $c = 1$.

(4.5) kan dus worden gereduceerd tot

$$(4.6) \quad \begin{cases} \dot{u}_2 = .04(1-u_2-u_3) - 10^4 u_2 u_3 - 3_{10}^7 u_2^2 \\ \dot{u}_3 = 3_{10}^7 u_2^2 \\ u_3 = u_3 = 0 \end{cases} \quad \begin{matrix} , \\ , t \geq 0 , \\ , t = 0 . \end{matrix}$$

De Jacobiaan van (4.6) heeft de vorm

$$J = \begin{bmatrix} -.04 - 6_{10}^7 u_2 - 10^4 u_3 & -.04 - 10^4 u_2 \\ & + 3_{10}^7 u_2^2 & 0 \end{bmatrix}$$

en heeft als eigenwaarden

$$\lambda_{1,2} = \frac{-b \pm \sqrt{b^2 - 4c}}{2} ,$$

waarbij

$$\begin{cases} b = .04 + 6_{10}^7 u_2 + 10^4 u_3 , \\ c = 6_{10}^7 u_2 (.04 + 10^4 u_1) . \end{cases}$$

We zien, dat als $4c \ll b^2$ - wat ook het geval zal blijken te zijn, omdat $u_2 \rightarrow 0$ en $u_3 \rightarrow 1$ -, voor de eigenwaarden geldt

$$\lambda_1 \approx -b , \quad \lambda_2 \approx -\frac{2c}{b} ,$$

zodat we dan met een stijve differentiaalvergelijking te maken hebben.

Aangezien de derde afgeleiden van u_2 en u_3 voor $t = 0$ bijzonder grote waarden hebben ($\dot{u}_1 = .04$, $\ddot{u}_2 = -.0016$, $\dot{u}_2 = -96_{10}^+3$, $\dot{u}_3 = \ddot{u}_3 = 0$, $\dot{u}_3 = +96_{10}^+3$, voor $t = 0$), moeten we om te beginnen zeer kleine stap-

lengtes nemen. We kozen daarom voor eta de formule

$$\text{eta} = (t+.01) * 10^{-3} + 10^{-6} * ||u||.$$

We integreerden (4.6) over het interval $[0,10]$ met .4 als referentie-punt. We kregen de volgende waarden van u_1 , u_2 en u_3

Tabel 4.3

t	u_1	u_2	u_3	k	l
.4	$.9850_{10^{-0}}$	$.3384_{10^{-4}}$	$.1489_{10^{-1}}$	19	94
10	$.8410_{10^{-0}}$	$.1620_{10^{-4}}$	$.1589_{10^{-0}}$	135	1252

In bovenstaande tabel geeft k het aantal integratiestappen op het tijdstip t aan en l het aantal evaluaties van de afgeleide (4.6). Het is duidelijk, dat er veel meer integratiestappen nodig waren om het interval $[0,0.4]$ te doorlopen dan om het interval $[0.4,10]$ te doorlopen. Voor de integratie van (4.6) over het interval $[\cdot 4,10]$ waren er evenwel veel meer evaluaties van de afgeleide nodig (nl. gemiddeld 10 per integratiestap) dan voor de integratie over het interval $[0,\cdot 4]$ (gemiddeld 5 per stap). De orde van nauwkeurigheid is aanvankelijk 3 en neemt dan snel af tot 1. De maximale geschatte fout is

$$\max_{k=1,\dots,135} ||\rho_k|| = .8802_{10^{-4}}.$$

De aanroep van ordervarying runge kutta was

```
k:= 0 ; t:= u[2]:= u[3]:= 0 ;
```

```
for te:= .4, 10 do
```

```
ordervarying runge kutta (t, te, 2, 3, u, sigma, derivative, k, 1.5,  
2, (t+.01) * 10-3, 10-6, eta, rho, output) ;
```


De procedure derivative was als volgt

```

procedure derivative (x,a) ; real x ; array a;
begin real a2 ;
    a2:= a[2] ;
    a[2]:= -a[3] * (104 * a2 + .04)
        -a2 * (.04 + 3*107 * a2) + .04 ;

    a[3]:= 3*107 * a2 * a2
end ;

```

sigma werd gegeven door middel van de funktieprocedure

```

real procedure sigma;
begin real b, c, u2;
    u2:= u[2];
    b:= .04 + 104 * u[3] + 6*107 * u2 ;
    c:= 2*107 * u2 * (.04 + 104 * u2) ;
    sigma:= (b + sqrt (b * b - c)) * .5
end ;

```

4.4 Een niet-autonome differentiaalvergelijking

Het beginwaardeprobleem

$$(4.7) \quad \begin{cases} \dot{u} = -e^t(u - \ln t) + \frac{1}{t}, & t \geq .01, \\ u = -\ln(100) & , t = .01 \end{cases}$$

heeft als exakte oplossing

$$\tilde{u} = \ln t .$$

Om twee redenen is (4.7) erg lastig om met behulp van een RK-schema te integreren.

- 1°. De spektraalradius van de Jacobiaan, e^t , is voor $t > 3$ erg groot en stijgt snel voor grotere t , zodat (4.7) dan een bijzonder stijve differentiaalvergelijking wordt.
- 2°. Het startpunt $t = .01$ ligt dicht in de buurt van het singuliere

punt $t = 0$, zodat voor $t \ll 1$ alle afgeleiden van u in absolute waarde groot zijn; daardoor geeft een derde of lagere orde consistent RK-schema alleen nauwkeurige resultaten, als τ klein is ;

Om deze moeilijkheden het hoofd te bieden, integreerden we (4.7) als volgt:

- a) op het interval $[.01, 1]$ werd τ door middel van een stapkeuzestrategie volgens [3,3.7] berekend;
daarvoor kozen we

$$\text{aeta} = 10^{-2} \quad , \quad \text{reta} = 10^{-4} ;$$

- b) op het interval $[1, 10]$ kozen we τ vast, waarbij τ slechts aan de stabiliteitsvoorwaarde was onderworpen;
we kozen als vaste staplente $\frac{1}{10}$, zodat

$$\tau = \min \left(\frac{1}{10}, 195e^{-t}, 10 - t \right) .$$

Een korte berekening toont aan, dat $\tau = \frac{1}{10}$, als

$$t > \ln(1950) \approx 7.57 .$$

Over het interval $[1, 7.6]$ wordt dus met een vaste staplengte .1 geïntegreerd en over het interval $[7.6, 10]$ wordt met de staplengte

$$\tau = \min (195e^{-t}, 10 - t) .$$

We vatten de resultaten samen in de volgende tabel

Tabel 4.4

Integratieinterval	$[.01, 1]$	$[1, 7.6]$	$[7.6, 10]$
ϵ_{\max}	$.190_{10}^{-3}$	$.304_{10}^{-3}$	$.923_{10}^{-4}$
ρ_{\max}	$.122_{10}^{-1}$	--	--
Aantal integratiestappen	36	66	102
Aantal evaluaties van de afgeleide	109	277	1018

In bovenstaande tabel is

$$\epsilon_{\max} = \max_{t_k \in \text{interval}} |\tilde{u}(t_k) - u_k|$$

en

$$\rho_{\max} = \max_{k=1, \dots, 36} ||\rho_k||.$$

De grootte $\tau \sigma(D)$ (die de orde van consistentie p en de graad n van (2.7) bepaalt) neemt langzaam toe tot 1 (bij de 50^e integratiestap) en stijgt dan snel tot 195 (bij de 100^e integratiestap) en blijft deze waarde dan behouden. p neemt af en n neemt toe volgens 2.2.

Een soortgelijk resultaat werd verkregen, als we als vaste staplengte op het interval $[1, 10]$.05 namen. We integreerden dan in feite over de intervallen $[.01, 1]$, $[1, 8.25]$ en $[8.25, 10]$

Tabel 4.5 Resultaten met $\tau = \min(\frac{1}{20}, 195e^{-t}, 10-t)$.

Integratieinterval	[.01,1]	[1,8.25]	[8.25,10]
ϵ_{\max}	$.190_{10^{-3}}$	$.154_{10^{-3}}$	$.701_{10^{-4}}$
ρ_{\max}	$.122_{10^{-1}}$	--	--
Aantal integratiestappen	36	145	93
Aantal evaluaties van de afgeleide	109	588	927

Literatuur

- [1] Van der Houwen, P.J., One step methods for solving linear initial value problems II; applications to stiff equations, TW 122/70, Mathematisch Centrum, 1970.
- [2] Van der Houwen, P.J., Stabilized Runge-Kutta methods with limited storage requirements, TW 123/71, Mathematisch Centrum, 1971.
- [3] Beentjes, P.A., Een ALGOL 60 versie van gestabiliseerde Runge Kutta methoden, NR 25/72, Mathematisch Centrum, 1972.
- [4] Dekker, T.J., ALGOL 60 procedures of numerical algebra I, M.C. tract 22, Mathematisch Centrum, 1968.
- [5] Bavinck, H., Jacobi Series and approximation, M.C. tract 39, Mathematisch Centrum, 1972.
- [6] Syllabus colloquium stijve differentiaalvergelijkingen en toepassingen in de biomathematica, Mathematisch Centrum (moet nog verschijnen).
- [7] Leon Lapidus and John H. Seinfeld, Numerical integration of ordinary differential equations, Academic Press, New York, London, 1971.